## REMARKS

Claims 1-4, 9-19, 71, 74-81, 86, 88, and 94-112 were pending and stand rejected. Claims 1-4, 9-19, 71, 74-81, 88, and 94-111 have been cancelled. Claims 86 and 112 have been amended. New claims 113-121 have been added. Claims 86 and 112-121 are pending upon entry of this amendment.

### Claims 1-4, 9-19, 71, 74-81, 88, and 94-111

Claims 1, 4, and 9-18 were rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Grinstein in view of Korein. Claim 2 was rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Grinstein in view of Korein and Walton. Claim 3 was rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Grinstein in view of Korein and Gagne. Claim 19 was rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Grinstein in view of Korein and Anderson. Claims 71, 74-75, 77-78, 94-96, 98-104, and 106-107 were rejected under 35 U.S.C. § 102(e) as allegedly being anticipated by Grinstein. Claims 76 and 79 were rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Grinstein in view of French. Claims 80-81 were rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Grinstein in view of Sowizral. Claims 88, 97, and 110 were rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Grinstein. Claim 105 was rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Grinstein in view of Gagne. Claims 108 and 111 were rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Grinstein in view of Walton. Claim 109 was rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Grinstein in view of Anderson.

Claims 1-4, 9-19, 71, 74-81, 88, and 94-111 have been cancelled.

Claim 86

Claim 86 was rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Grinstein in view of Unuma. Applicants respectfully traverse in view of the amended claim. As amended, claim 86 recites:

In a computer-implemented animation system, a method for animating an object, the method comprising:

> receiving an input specifying an Align to Motion behavior, the Align to Motion behavior indicating how to change a value of a rotation parameter of the object over time based on a motion path of the object such that the rotation parameter is not changed if the motion path is straight;
>
> animating the object by changing the value of the rotation parameter of the object over time according to the Align to Motion behavior; and
>
> outputting the animated object;

wherein the Align to Motion behavior can be configured regarding:

> a spring tension parameter, which determines how quickly the object's rotation changes based on a change in the object's motion path;
>
> an axis parameter, which determines whether the object's rotation is based on an X value of the object's position or a Y value of the object's position; and
>
> a drag parameter, which determines whether or not the object's change in rotation overshoots a new direction of the object.

Claim 86 recites, in part, "receiving an input specifying an Align to Motion behavior, the Align to Motion behavior indicating how to change a value of a rotation parameter of the object over time based on a motion path of the object such that the rotation parameter is not changed if the motion path is straight … wherein the Align to Motion behavior can be configured regarding: … a drag parameter, which determines whether or not the object's change in rotation overshoots a new direction of the object."

As described in the pending application (¶¶580-589[1]), the Align to Motion behavior is meant to be applied to a moving object (i.e., an object whose position parameter is changing over time) (¶581). This behavior changes the rotation of the object to match changes made to the object's direction along its motion path (¶581). The Align to Motion behavior can be used, for example, to cause an object to face the direction in which it is moving. Unlike the Snap

Alignment to Motion behavior, which produces absolute changes in rotation that precisely match changes in direction, the Align to Motion behavior has a springy effect (¶582) due to the Spring Tension parameter (¶587) and the Drag parameter (¶588).

Applicants agree with the Examiner that Grinstein does not disclose, teach, or suggest "wherein the behavior comprises … an Align to Motion behavior … which changes a rotation of the object based on a motion path of the object such that the rotation is not changed if the motion path is straight" (Detailed Action, page 24). It follows that Grinstein also does not disclose, teach, or suggest the claimed element "receiving an input specifying an Align to Motion behavior, the Align to Motion behavior indicating how to change a value of a rotation parameter of the object over time based on a motion path of the object such that the rotation parameter is not changed if the motion path is straight … wherein the Align to Motion behavior can be configured regarding: … a drag parameter, which determines whether or not the object's change in rotation overshoots a new direction of the object."

Unuma does not remedy this deficiency. Unuma discusses a transit point specifying unit and a moving direction controller (¶133; FIG. 17). The transit point specifying unit specifies transit points that are connected to each other with a curve so as to create a moving route designated by a position and a curve (¶133; FIG. 18). Then, the moving direction controller rotates the object so that the front side of the object is oriented to a direction of a tangent of the curve at a position of the object moving on the curve (¶133).

Claim 86 recites, in part, "wherein the Align to Motion behavior can be configured regarding: … a drag parameter, which determines <u>whether or not the object's change in rotation overshoots a new direction of the object</u>" (emphasis added). In Unuma, the object's rotation is

---

[1] Paragraph citations are to the application as published.

always equal to the tangent of the object's motion path. Thus, the rotation cannot be configured regarding whether or not the object's change in rotation overshoots a new direction of the object.

It follows that Unuma does not disclose, teach, or suggest "wherein the Align to Motion behavior can be configured regarding: … a drag parameter, which determines whether or not the object's change in rotation overshoots a new direction of the object."

Thus, neither Grinstein nor Unuma, alone or in combination, discloses, teaches, or suggests the claimed element "receiving an input specifying an Align to Motion behavior, the Align to Motion behavior indicating how to change a value of a rotation parameter of the object over time based on a motion path of the object such that the rotation parameter is not changed if the motion path is straight … wherein the Align to Motion behavior can be configured regarding: … a drag parameter, which determines whether or not the object's change in rotation overshoots a new direction of the object."

Therefore, claim 86 is patentable over Grinstein and Unuma, alone and in combination.

Claim 117 recites similar language and is patentable for at least the same reasons.

Claim 112

Claim 112 was rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Grinstein in view of Anderson. Applicants respectfully traverse in view of the amended claim. As amended, claim 112 recites:

In a computer-implemented animation system, a method for animating an object, the method comprising:

receiving an input specifying an Attracted To behavior, the Attracted To behavior indicating how to change a value of a position parameter of the object over time based on a position of a second object while not affecting the position of the second object;

animating the object by changing the value of the position parameter of the object over time according to the Attracted To behavior; and

outputting the animated object;

wherein the Attracted To behavior can be modified using:

a falloff rate parameter, which determines a rate of acceleration with which the object moves towards the second object;

an influence parameter, which determines an area of influence, the area of influence determining whether the object is affected by the Attracted To behavior;

a falloff type parameter, which determines whether a distance defined by the influence parameter falls off linearly or exponentially;

a strength parameter, which determines a speed at which the object moves towards the second object; and

a drag parameter, which determines whether the object overshoots the second object.

Claim 112 recites, in part, "receiving an input specifying an Attracted To behavior, the Attracted To behavior indicating how to change a value of a position parameter of the object over time based on a position of a second object while not affecting the position of the second object … wherein the Attracted To behavior can be configured using: … an influence parameter, which determines an area of influence, the area of influence determining whether the object is affected by the Attracted To behavior; a falloff type parameter, which determines whether a distance defined by the influence parameter falls off linearly or exponentially."

As described in the pending application (¶¶590-604; FIG. 48), the Attracted To behavior affects an object's Position parameter (¶592). An object with the Attracted To behavior (the "attracted object") moves towards a single specified object, the object of attraction (the "attracting object") (¶592). Additional parameters allow a user to adjust the area of influence that defines how close an object needs to be to move towards the object of attraction and how strongly it is attracted (¶592).

Applicants agree with the Examiner that Grinstein does not disclose, teach, or suggest "wherein the behavior comprises … a Drift Attracted To behavior, which changes a position of the object based on a position of a second object while not affecting the position of the second object" (Detailed Action, page 29). It follows that Grinstein also does not disclose, teach, or suggest the claimed element "receiving an input specifying an Attracted To behavior, the Attracted To behavior indicating how to change a value of a position parameter of the object

over time based on a position of a second object while not affecting the position of the second object … wherein the Attracted To behavior can be configured using: … an influence parameter, which determines an area of influence, the area of influence determining whether the object is affected by the Attracted To behavior; a falloff type parameter, which determines whether a distance defined by the influence parameter falls off linearly or exponentially."

Anderson does not remedy this deficiency. Anderson discusses animation controls that permit the effect of a simulation to be overridden for cloth and soft body materials (abstract). A character 10 includes an object 14 (e.g., a model of a person) wearing a garment 12 (e.g., a tie) (3:44-45). "Shape tacks" control the local shape of objects such as cloth and soft body materials (abstract). The shape can be weighted and time varying (abstract).

The Examiner argued that Anderson's cloth (e.g., a necktie) corresponds to the claimed element "object" (attracted object) and that Anderson's underlying object (e.g., person model) corresponds to the claimed element "second object" (attracting object) (Detailed Action, p. 29). Specifically, the Examiner cited Anderson's position tacks (4:55-5:33),which make use of a weight field that represents the degree to which a reference motion (e.g., a motion of the person model) will be followed by a particular part of the simulation object such as a tie (Detailed Action, p. 29).

Claim 112 recites, in part, "wherein the Attracted To behavior can be configured using: … an influence parameter, which determines <u>an area of influence, the area of influence determining whether the object is affected by the Attracted To behavior</u>; a falloff type parameter, which determines <u>whether a distance defined by the influence parameter falls off linearly or exponentially</u>" (emphasis added). Anderson discloses neither an influence parameter nor a falloff type parameter that determines whether a distance defined by the influence parameter falls off linearly or exponentially.

It follows that Anderson does not disclose, teach, or suggest “wherein the Attracted To behavior can be configured using: … an influence parameter, which determines an area of influence, the area of influence determining whether the object is affected by the Attracted To behavior; a falloff type parameter, which determines whether a distance defined by the influence parameter falls off linearly or exponentially.”

Thus, neither Grinstein nor Anderson, alone or in combination, discloses, teaches, or suggests the claimed element “receiving an input specifying an Attracted To behavior, the Attracted To behavior indicating how to change a value of a position parameter of the object over time based on a position of a second object while not affecting the position of the second object … wherein the Attracted To behavior can be configured using: … an influence parameter, which determines an area of influence, the area of influence determining whether the object is affected by the Attracted To behavior; a falloff type parameter, which determines whether a distance defined by the influence parameter falls off linearly or exponentially.”

Therefore, claim 112 is patentable over Grinstein and Anderson, alone and in combination.

Claim 118 recites similar language and is patentable for at least the same reasons.

<u>New Claims 113-121</u>

Although claims 113-121 have not been rejected, Applicants respectfully note the following:

Claim 113 recites:

In a computer-implemented animation system, a method for animating an object, the method comprising:

receiving an input specifying a Grow/Shrink behavior, the Grow/Shrink behavior indicating how to change a value of a scale parameter of the object over time by

either changing a size of the object by a steady number of pixels per second or changing the object's size from an original size to a final size;

animating the object by changing the value of the scale parameter of the object over time according to the Grow/Shrink behavior; and

outputting the animated object;

wherein the Grow/Shrink behavior can be configured regarding:

if the object's size is changing by a steady number of pixels per second, a first number of horizontal pixels and a second number of vertical pixels, wherein the first number is added to the object's horizontal size each second, and wherein the second number is added to the object's vertical size each second;

if the object's size is changing from the original size to the final size, a first number of horizontal pixels and a second number of vertical pixels, wherein the first number represents horizontal pixels in the final size, and where the second number represents vertical pixels in the final size; and

an acceleration with which the object's size changes over time.

As described in the pending application (¶¶435-445; FIG. 39), the Grow/Shrink behavior affects an object's Scale parameter (¶436). The Grow/Shrink behavior is used to animate the scale of an object, enlarging or reducing the object's size over time at a speed defined by the Scale Rate or Scale To parameter (¶436). The Grow/Shrink effect begins at the object's original size (¶436).

Claim 113 recites, in part, "receiving an input specifying a Grow/Shrink behavior, the Grow/Shrink behavior indicating how to change a value of a scale parameter of the object over time by either changing a size of the object by a steady number of pixels per second or changing the object's size from an original size to a final size … wherein the Grow/Shrink behavior can be configured regarding: … an acceleration with which the object's size changes over time."

None of the cited references discloses, teaches, or suggests this claimed element.

Therefore, claim 113 is patentable over the cited references, alone and in combination.

Claim 119 recites similar language and is patentable for at least the same reasons. Claim 114 depends from claim 113 and is patentable for at least the same reasons.

Claim 115 recites:

In a computer-implemented animation system, a method for animating an object, the method comprising:

receiving an input specifying an Orbit Around behavior, the Orbit Around behavior indicating how to change a value of a position parameter of the object over time based on a position of a second object while not affecting the position of the second object;

animating the object by changing the value of the position parameter of the object over time according to the Orbit Around behavior; and

outputting the animated object;

wherein the Orbit Around behavior can be configured regarding:

a falloff rate parameter, which determines a rate of acceleration with which the object moves around the second object;

an influence parameter, which determines an area of influence, the area of influence determining whether the object is affected by the Orbit Around behavior;

a falloff type parameter, which determines whether a distance defined by the influence parameter falls off linearly or exponentially; and

a strength parameter, which determines a speed at which the object moves around the second object.

As described in the pending application (¶¶675-688; FIGS. 51-52), the Orbit Around behavior affects an object's Position parameter (¶676). Similar to the Attracted To behavior, the Orbit Around behavior's default parameter settings cause an object to orbit around another object in a circle (¶676). FIG. 51 illustrates a first object orbiting around a second object and an orbit motion path of the first object (¶676).

Claim 115 recites, in part, "receiving an input specifying an Orbit Around behavior, the Orbit Around behavior indicating how to change a value of a position parameter of the object over time based on a position of a second object while not affecting the position of the second object … wherein the Orbit Around behavior can be configured regarding: … an influence parameter, which determines an area of influence, the area of influence determining whether the object is affected by the Orbit Around behavior; a falloff type parameter, which determines whether a distance defined by the influence parameter falls off linearly or exponentially."

None of the cited references discloses, teaches, or suggests this claimed element.

Therefore, claim 115 is patentable over the cited references, alone and in combination.

Claim 120 recites similar language and is patentable for at least the same reasons.

Claim 116 recites:

In a computer-implemented animation system, a method for animating an object, the method comprising:

receiving an input specifying a Random Motion behavior, the Random Motion behavior indicating how to change a value of a position parameter of the object over time based on a random motion path;

animating the object by changing the value of the position parameter of the object over time according to the Random Motion behavior; and

outputting the animated object;

wherein the Random Motion behavior can be configured regarding:

an amount parameter, which determines a length of the motion path;

a frequency parameter, which determines a crookedness of the motion path;

a noisiness parameter, which determines a level of jaggedness along the motion path; and

a drag parameter, which determines a speed at which the object moves along the motion path.

As described in the pending application (¶¶689-700; FIGS. 53-56), the Random Motion behavior affects an object's Position parameter (¶690). If a user applies the Random Motion behavior to an object, the behavior animates the position the object and makes the object move around the Canvas along a random path (¶690). FIG. 53 illustrates an object and a Random Motion motion path (¶690).

Claim 116 recites, in part, "receiving an input specifying a Random Motion behavior, the Random Motion behavior indicating how to change a value of a position parameter of the object over time based on a random motion path … wherein the Random Motion behavior can be configured regarding: … a frequency parameter, which determines a crookedness of the motion path; a noisiness parameter, which determines a level of jaggedness along the motion path."

None of the cited references discloses, teaches, or suggests this claimed element.

Therefore, claim 116 is patentable over the cited references, alone and in combination.

Claim 121 recites similar language and is patentable for at least the same reasons.

Applicants respectfully submit that the pending claims are allowable over the cited art of record and request that the Examiner allow this case. The Examiner is invited to contact the undersigned in order to advance the prosecution of this application.

Respectfully submitted,
GREGORY E. NILES, ET AL.

Dated: January 19, 2009

By: /Sabra-Anne R. Truesdale/

Sabra-Anne R. Truesdale, Reg. No. 55,687
Fenwick & West LLP
Silicon Valley Center
801 California Street
Mountain View, CA 94041
Tel.: (650) 335-7187
Fax.: (650) 938-5200